

Distributed Adaptive Filtering With Reduced Communication Load

Ihsan Utlü and Suleyman S. Kozat

Department of Electrical and Electronics Engineering

Bilkent University, Bilkent, Ankara 06800, Turkey

Email: {utlu,kozat}@ee.bilkent.edu.tr

Abstract—We propose novel algorithms for distributed processing in applications constrained by available communication resources, using diffusion strategies that achieve up to three orders-of-magnitude reduction in communication load on the network, while delivering equal performance with respect to the state of the art. After computation of local estimates, the information is diffused among processing elements (or nodes) non-uniformly in time by conditioning the information transfer on level-crossings of the diffused parameter, resulting in a greatly reduced communication requirement. We provide the mean stability analysis of our algorithms, and illustrate the gain in communication efficiency compared to other reduced-communication distributed estimation schemes.

I. INTRODUCTION

Distributed networks and signal processing algorithms have been a subject of growing interest in recent years, in view of their desirable characteristics such as intrinsic robustness and scalability [1], allowing for enhanced efficiency and performance in a large class of applications including wireless sensor networks, environmental surveillance, target localization, and distributed resource allocation [2], [3]. However, successful implementation of such applications depends on a substantial amount of communication resources. As an example, in smart grid applications, measurement units operating with high frequency put the communication infrastructure of the grid under significant pressure [4]. This calls for resource-efficient, event-triggered distributed estimation solutions that incorporate event-driven communication. To this end, in this paper, we construct distributed architectures that have a significantly reduced communication load without compromising performance. We achieve this by introducing novel event triggered communication architectures over distributed networks.

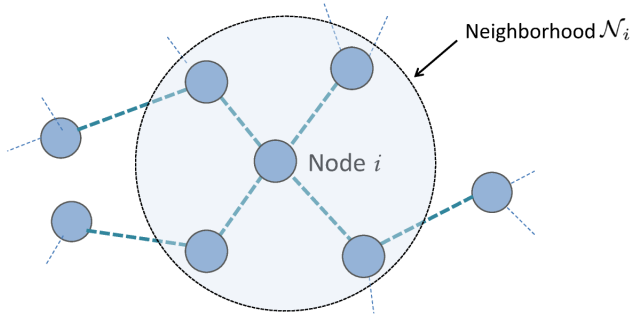
In a distributed processing framework, a group of measurement-capable agents, termed nodes, in a network cooperate with one another in order to estimate an unknown common phenomenon [5]. Among the different approaches, we specifically consider diffusion-based protocols that exploit the spatial diversity of the network by restricting information sharing to neighboring nodes, without considering any central processing unit or a fusion center [1], [5]. Diffusion protocols provide an inherently scalable data processing framework that is resilient to changes in network topology such as link failures as well as changes in the statistical properties of the unknown phenomenon that is measured [5]. However, the requirement for all nodes to exchange their current estimates with their neighbors at each iteration places a heavy burden on the available communication resources [6].

Here, we propose novel event-triggered distributed estimation algorithms for communication-constrained applications that achieve up to three orders-of-magnitude reduction in the communication load over the network. We achieve this by leveraging the uneven distribution of the events over time to efficiently reduce the communication load in real life applications. In particular, we condition an information exchange between the neighboring nodes on the level-crossings of the diffused parameter [7], unlike using a fixed rate of diffusion, cf. [1], [5]. Furthermore, we show that it is sufficient to only diffuse the information indicating the direction of the change in the levels, which can be handled using only a single bit for a slowly-varying parameter.

Reduced communication diffusion is extensively studied in the signal processing literature [6], [8]–[11]. In [6], [8], [9], the authors restrict the number of active links between neighbors using a probabilistic framework, or by adaptively choosing a single link of communication for each node. In [10], local estimates are randomly projected, and the information transfer between the nodes is reduced to a single bit. In [11], only certain dimensions of the parameter vector are transmitted. On the other hand, in this paper, we reduce the communication load down to only a single bit or a couple of bits, unlike [6], [8], [9], [11], in which authors diffuse parameters in full precision. Furthermore, we regulate the frequency of information exchange depending on the rate of change of the parameter, unlike [10] where the authors transfer information at each single time instant.

Our main contributions are as follows. We introduce algorithms for distributed estimation that i) significantly reduce the communication load on the network, ii) while continuing to provide equal performance with the state of the art. We also perform the mean-stability analysis of our algorithms. Through numerical examples, we show that our algorithms achieve up to three orders-of-magnitude reduction in the communication load over the network.

The paper is organized as follows: In Section II, we introduce the distributed estimation framework and discuss the adapt-then-combine (ATC) diffusion strategy. We further detail our algorithms in Section III, where we formulate the level-triggered distributed estimation algorithm. In Section IV, we present the algorithmic description of the proposed scheme. In Section V, we provide the mean stability analysis of the proposed distributed adaptive filter and state the conditions for stability. We provide experimental verification of the algorithm in Section VI, and concluding remarks in Section VII.

Fig. 1. An example network with N nodes.

II. PROBLEM DESCRIPTION

Consider a network with N nodes that are distributed spatially as shown in Fig. 1. Each node sequentially observes a noise-corrupted transformation of an unknown parameter \mathbf{w}_o through a linear model

$$d_{i,t} = \mathbf{u}_{i,t}^T \mathbf{w}_o + v_{i,t}, \quad i = 1, \dots, N \quad (1)$$

and diffuses information to its neighboring nodes¹, where $\mathbf{w}_o \in \mathbb{R}^M$ is the unknown phenomenon, with $\mathbf{u}_{i,t}$ and $v_{i,t}$ representing the regressor and noise processes, respectively. The additive observation noise $v_{i,t}$ and the regressor $\mathbf{u}_{i,t}$ are assumed to be temporally and spatially independent, and independent of one another, with $\mathbf{\Lambda}_i \triangleq E[\mathbf{u}_{i,t} \mathbf{u}_{i,t}^T] = \sigma_{u,i}^2 \mathbf{I}_M$, $E[v_{i,t}^2] = \sigma_{v,i}^2$.

At each node i , an adaptive estimation algorithm is used to recover \mathbf{w}_o such as a stochastic gradient approach [12] as

$$\boldsymbol{\varphi}_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t} \mathbf{u}_{i,t}^T) \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} d_{i,t},$$

$\mu_i > 0$, where $\mathbf{w}_{i,t}$ represents the current estimate and $\boldsymbol{\varphi}_{i,t+1}$ is the updated intermediary estimate after the new observation. In this paper we consider the adapt-then-combine (ATC) diffusion strategy as an example [1] where, at each node i , the final estimate is constructed as

$$\mathbf{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} p_{i,j} \boldsymbol{\varphi}_{j,t+1}, \quad (2)$$

after the intermediary estimates of the neighboring nodes arrive to the node i , where \mathcal{N}_i is the neighborhood of the node i , $\mathbf{P} = [p_{i,j}]$ is the network matrix comprised of the combination weights $\sum_{j=1}^N p_{i,j} = 1$ and $p_{i,j} \geq 0$.

III. DISTRIBUTED ESTIMATION WITH LEVEL TRIGGERED SAMPLING

The well-known ATC full diffusion scheme (2) requires all nodes in the network to communicate their current estimates (i) in their entirety, and (ii) at a fixed rate to all their neighboring nodes [1]. We propose a new scheme, which

¹We represent vectors (matrices) by bold lower (upper) case letters. For a vector \mathbf{a} (a matrix \mathbf{A}), \mathbf{a}^T (\mathbf{A}^T) is the transpose. $\|\mathbf{a}\|$ represents the Euclidean norm. The $\text{diag}\{\mathbf{A}\}$ returns a new matrix with only the main diagonal of \mathbf{A} while $\text{bdiag}\{\mathbf{A}_1, \dots\}$ puts its arguments on the diagonal blocks of the new matrix. $\text{col}\{\mathbf{a}_1, \dots, \mathbf{a}_N\}$ produces a column vector formed by column-wise stacking its arguments on top of one another. \mathbf{I}_M represents the $M \times M$ identity matrix. \otimes stands for the Kronecker product.

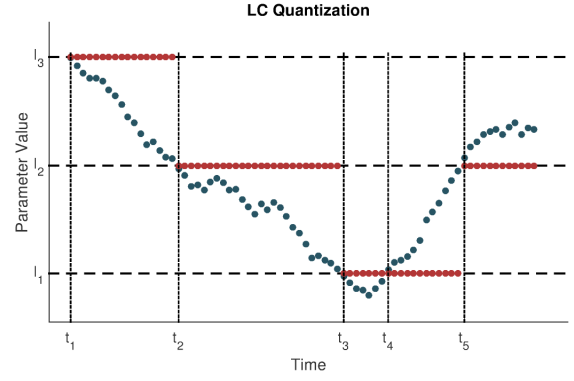


Fig. 2. Illustration of the operation of the LC quantizer.

achieves an increased communication efficiency by conditioning the diffusion of information on the *trigger of an event*, instead of relying on a fixed rate of diffusion. Our approach considerably reduces the load on communication resources since only “significant changes” in the diffused parameter, e.g., an abrupt change in the local estimate, are conveyed based on the particular realization of the signal.

To clarify the framework, we consider the diffusion of a scalar parameter $\xi_{i,t}$ from a given node i to a neighboring node j . As an example, this information can be a single component of the estimates [11], or the error associated with an additional estimation layer [10]. In our distributed framework, due to communication constraints, a quantized version of the original parameter, $\xi_{i,t}^q$ is shared. We aim to form a quantization scheme, which guarantees that $\xi_{i,t}$ and $\xi_{i,t}^q$ are approximately equal to each other for all t , while at the same time keeping the load on communication resources relatively small.

To solve this problem, we propose an event-triggered communication algorithm where, as the event-triggered approach, we specifically use level crossing (LC) quantization [7]. To clarify the framework, suppose we have a discrete time signal $\xi_{i,t}$ as shown in Fig. 2 that represents the information to be communicated from the node i to the node j , e.g., the estimated parameter, or the estimation error. In conventional quantization, at each time instant, we sample and quantize this parameter. On the other hand, in the LC quantization, we consider a set of levels $\mathcal{S} \triangleq \{l_1, \dots, l_K\}$, which is illustrated in Fig. 2. At each discrete time index t , the node i checks whether a level-crossing has occurred on $\xi_{i,t}$. When the parameter $\xi_{i,t}$ crosses a level $l_{i,t}$, i.e.,

$$(\xi_{i,t-1} - l_{i,t})(\xi_{i,t} - l_{i,t}) < 0 \quad \text{for some } l_{i,t} \in \mathcal{S},$$

the node i transmits information to its neighboring nodes. For example, this information can be the direction of the level-crossing [7]. A neighboring node j uses this received information to form an estimate $\xi_{i,t}^q$ for $\xi_{i,t}$.

If there is an information transfer by the node i at time t , the receiving node j estimates the parameter as the level through which a level crossing has occurred:

$$\xi_{i,t}^q = l_{i,t}. \quad (3)$$

For the time instants when the node i is silent, the node j infers that no significant change in the parameter has taken place,

and uses the estimated parameter value from the previous time instant:

$$\xi_{i,t}^q = \xi_{i,t-1}^q. \quad (4)$$

We note that the set of levels \mathcal{S} is known by all nodes in the network. Hence, as the diffused information, it is sufficient for the node i to only convey how $\xi_{i,t}^q$ changes compared to the previously-crossed level $\xi_{i,t-1}^q$. In particular, we note the following two cases: In the first case, the parameter $\xi_{i,t}^q$ changes slowly enough such that a crossing through multiple levels do not occur, so that the node i only needs to indicate the *direction* of the change in levels, which we represent using a single bit. In the second case, we may have multiple crossings where we directly code with a flag bit the full location information of the new level value $\xi_{i,t}^q$ using $\lceil \log_2(K) \rceil + 1$ bits. As shown, this approach significantly lowers the amount of communication while maintaining estimation performance.

IV. ALGORITHM DESCRIPTION

In this section, we present the full algorithmic description of the proposed diffusion scheme with the level-crossing quantization [7]. At time t , a given node i in the network makes the scalar observation $d_{i,t}$ through the linear model $d_{i,t} = \mathbf{u}_{i,t}^T \mathbf{w}_o + v_{i,t}$, which is then used to update its intermediary local estimate using the LMS adaptation

$$\varphi_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t} \mathbf{u}_{i,t}^T) \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} d_{i,t}.$$

Due to the quantized communication framework, a neighboring node j does not have access to the true value of the parameter $\varphi_{i,t+1}$, which has M entries. As such, based on the limited information it receives from the node i , the node j tries to estimate this parameter as the M -entry vector $\varphi_{j,t+1}^q$. Specifically, in the LC quantization, the node j receives information about how the current values of the *entries* of the parameter $\varphi_{i,t+1}$ have changed relative to the most recent estimate the node j has access to, namely $\varphi_{i,t}^q$. In order to provide this information, the node i also keeps a record of the past estimated parameter values $\{\varphi_i^q(k)\}_{k=1}^t$ that the neighboring nodes have related to its true $\{\varphi_i(k)\}_{k=1}^t$. The node i uses the most recent entry in this record, $\varphi_{i,t}^q$, as a reference and diffuses information to the neighboring nodes j indicating how the current estimate $\varphi_{i,t+1}$ compares to this reference on a per-entry basis. In particular, the node i makes this comparison by checking for a level crossing between corresponding entries of the two vector quantities $\varphi_{i,t}^q$ and $\varphi_{i,t+1}$. If there is a level crossing on an entry, the node i transmits information to its neighbors through a channel frequency allocated to this particular entry. If there is a single level-crossing, this information indicates the direction of the level crossing; otherwise, the transmitted information directly specifies the location of the new level. A neighboring node j then constructs the estimate $\varphi_{j,t+1}^q$ using (3) or (4) on a per-entry basis, depending on whether the node i diffuses information or not, respectively, at time t .

While diffusing information related to its own local estimate, the node i also receives information from the neighboring nodes j representing their local estimates $\varphi_{j,t+1}$. For each neighboring node j , the node i uses this diffused information

to reconstruct $\varphi_{j,t+1}^q$ using (3) or (4). The final estimate $\mathbf{w}_{i,t+1}$ is then constructed using the combination

$$\mathbf{w}_{i,t+1} = p_{i,i} \varphi_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j} \varphi_{j,t+1}^q.$$

Remark: In order to keep the presentation clear, we illustrate the special case of $M = 1$ of the proposed algorithm in Algorithm 1, which can be generalized to arbitrary M in a straightforward manner.

Remark: We note that an alternative approach to dealing with the $M > 1$ case is to have the nodes in the network transmit only a certain entry of their intermediary estimates $\varphi_{i,t}$. As an example, in this case, the nodes can cycle through different entries across time in a round-robin fashion. The non-communicated entries are replaced by the corresponding entries in the local intermediary estimate [11]. This approach is explored in the Experiments section.

V. MEAN STABILITY ANALYSIS

To continue with the stability analysis of the proposed scheme, we assume that the regressors $\mathbf{u}_{i,t}$ are temporally and spatially independent, zero mean and white, with covariance matrix $\mathbf{\Lambda}_i \triangleq E[\mathbf{u}_{i,t} \mathbf{u}_{i,t}^T] = \sigma_{u,i}^2 \mathbf{I}_M$. The observation $d_{i,t}$ at node i is assumed to follow a linear model of the form

$$d_{i,t} = \mathbf{u}_{i,t}^T \mathbf{w}_o + v_{i,t}, \quad (5)$$

where $\{v_{i,t}\}_{t \geq 1}$ is a white Gaussian noise process with variance $\sigma_{v,i}^2$, independent of $\{\mathbf{u}_{j,t}\}_{t \geq 1} \forall i, j$.

In our proposed level-triggered estimation framework, at each node i , the diffusion LMS update for the ATC strategy take the form

$$\varphi_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t} \mathbf{u}_{i,t}^T) \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} d_{i,t}, \quad (6)$$

$$\mathbf{w}_{i,t+1} = p_{i,i} \varphi_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j} \varphi_{j,t+1}^q, \quad (7)$$

where the combination matrix \mathbf{P} is taken to be stochastic, with its rows summing up to unity. We rewrite the expressions (6) and (7) as

$$\varphi_{i,t+1} = (\mathbf{I}_M - \mu_i \mathbf{u}_{i,t} \mathbf{u}_{i,t}^T) \mathbf{w}_{i,t} + \mu_i \mathbf{u}_{i,t} d_{i,t}, \quad (8)$$

$$\mathbf{w}_{i,t+1} = \sum_{j \in \mathcal{N}_i} p_{i,j} \varphi_{j,t+1} - \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j} \alpha_{j,t+1}, \quad (9)$$

by defining the quantization error for node j

$$\alpha_{j,t} \triangleq \varphi_{j,t} - \varphi_{j,t}^q.$$

We represent the diffusion update over the network \mathcal{N} in state-space form by introducing the following global quantities:

$$\begin{aligned} \mathbf{d}_t &\triangleq \text{col}\{d_{1,t}, \dots, d_{N,t}\} & \mathbf{v}_t &\triangleq \text{col}\{v_{1,t}, \dots, v_{N,t}\} \\ \mathbf{w}_o &\triangleq \text{col}\{\mathbf{w}_o, \dots, \mathbf{w}_o\} & \mathbf{U}_t &\triangleq \text{bdiag}\{\mathbf{u}_{1,t}, \dots, \mathbf{u}_{N,t}\} \\ \mathbf{M} &\triangleq \text{bdiag}\{\mu_1 \mathbf{I}_M, \dots, \mu_N \mathbf{I}_M\} & \mathbf{w}_t &\triangleq \text{col}\{\mathbf{w}_{1,t}, \dots, \mathbf{w}_{N,t}\} \\ \boldsymbol{\varphi}_t &\triangleq \text{col}\{\boldsymbol{\varphi}_{1,t}, \dots, \boldsymbol{\varphi}_{N,t}\} & \boldsymbol{\varphi}_t^q &\triangleq \text{col}\{\boldsymbol{\varphi}_{1,t}^q, \dots, \boldsymbol{\varphi}_{N,t}^q\} \\ \boldsymbol{\alpha}_t &\triangleq \text{col}\{\boldsymbol{\alpha}_{1,t}, \dots, \boldsymbol{\alpha}_{N,t}\} & \mathbf{G} &\triangleq \mathbf{P} \otimes \mathbf{I}_M \\ \mathbf{G}_C &\triangleq (\mathbf{P} - \text{diag}\{\mathbf{P}\}) \otimes \mathbf{I}_M \end{aligned}$$

Algorithm 1 ATC Diffusion LMS with the LC Quantization, $M=1$

```

1: for  $i = 1$  to  $N$  do
    Initialization:
2:    $w_{i,0} = \varphi_{i,0}^q = 0$ 
3: end for
4: for  $t \geq 0$  do
5:   for  $i = 1$  to  $N$  do
    Local adaptation:
6:    $\varphi_{i,t+1} = (1 - \mu_i u_{i,t}^2) w_{i,t} + \mu_i u_{i,t} d_{i,t}$ 
    Check for level crossing:
7:   if  $\exists l_{i,t} \in \mathcal{S}$  such that
     $(\varphi_{i,t}^q - l_{i,t})(\varphi_{i,t+1} - l_{i,t}) < 0$  then
8:     if The crossing is to an adjacent level then
9:       Diffuse the direction of the crossing
10:    else
11:      Diffuse the location of the new level
12:    end if
13:    Locally store  $\varphi_{i,t+1}^q = l_{i,t}$  in record
14:  else
15:    Remain silent
16:    Locally set  $\varphi_{i,t+1}^q = \varphi_{i,t}^q$ 
17:  end if
    Reconstruction:
18:  for all  $j \in \mathcal{N}_i \setminus \{i\}$  do
19:    if node  $j$  is silent then
20:      Reconstruct as  $\varphi_{j,t+1}^q = \varphi_{j,t}^q$ 
21:    else
22:      Reconstruct  $\varphi_{j,t+1}^q$  using the diffused
        information
23:    end if
24:  end for
    Combination:
25:   $w_{i,t+1} = p_{i,i} \varphi_{i,t+1} + \sum_{j \in \mathcal{N}_i \setminus \{i\}} p_{i,j} \varphi_{j,t+1}^q$ 
26: end for
27: end for

```

Using the above-defined quantities, the diffusion updates (8), (9) take the following global state-space form:

$$\varphi_{t+1} = (\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t \mathbf{U}_t^T) \mathbf{w}_t + \mathbf{M}\mathbf{U}_t \mathbf{d}_t, \quad (10)$$

$$\mathbf{w}_{t+1} = \mathbf{G}\varphi_{t+1} - \mathbf{G}_C \alpha_{t+1}. \quad (11)$$

Similarly, the data model (5) can be expressed in terms of the global quantities as

$$\mathbf{d}_t = \mathbf{U}_t^T \mathbf{w}_o + \mathbf{v}_t. \quad (12)$$

To facilitate the mean stability analysis, we define the global deviation parameters

$$\begin{aligned} \tilde{\mathbf{w}}_t &\triangleq \mathbf{w}_o - \mathbf{w}_t, \\ \tilde{\varphi}_t &\triangleq \mathbf{w}_o - \varphi_t. \end{aligned}$$

After substituting (12) and subtracting both sides of (10), (11) from \mathbf{w}_o , the diffusion updates in terms of the deviation parameters take the following form:

$$\tilde{\varphi}_{t+1} = (\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t \mathbf{U}_t^T) \tilde{\mathbf{w}}_t - \mathbf{M}\mathbf{U}_t \mathbf{v}_t, \quad (13)$$

$$\tilde{\mathbf{w}}_{t+1} = \mathbf{G}\tilde{\varphi}_{t+1} + \mathbf{G}_C \alpha_{t+1}, \quad (14)$$

where we have used the relation $\mathbf{G}\mathbf{w}_o = \mathbf{w}_o$, which results from the stochastic nature of \mathbf{P} .

The expressions (13), (14) can be expressed compactly as

$$\begin{aligned} \tilde{\mathbf{w}}_{t+1} &= \mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{U}_t \mathbf{U}_t^T) \tilde{\mathbf{w}}_t \\ &\quad - \mathbf{G}\mathbf{M}\mathbf{U}_t \mathbf{v}_t + \mathbf{G}_C \alpha_{t+1}. \end{aligned} \quad (15)$$

Assumption: The quantization error over the network α_t has zero mean. This is a reasonable assumption for the analysis of quantization effects [12]. The applicability of the assumption is verified by our experiments in Section VI.

Taking expectations of both sides of (15) yields

$$\mathbf{E}[\tilde{\mathbf{w}}_{t+1}] = \mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda}) \mathbf{E}[\tilde{\mathbf{w}}_t], \quad (16)$$

where $\mathbf{\Lambda} \triangleq \text{bdiag}\{\mathbf{\Lambda}_1, \dots, \mathbf{\Lambda}_N\}$ is block diagonal.

For mean stability and asymptotic unbiasedness of the distributed filter (6)-(7), we require that the spectral radius $|\mathbf{G}(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda})| < 1$, which, noting that \mathbf{G} is stochastic with nonnegative entries, is equivalent to requiring

$$|(\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda})| < 1, \quad (17)$$

by Lemma 1 of [1]. Noting that the eigenvalues of the block diagonal matrix $\mathbf{I}_{MN} - \mathbf{M}\mathbf{\Lambda}$ is the union of the eigenvalues of its individual blocks $\mathbf{I}_M - \mu_i \mathbf{\Lambda}_i$ where $\mathbf{\Lambda}_i = \sigma_{u,i}^2 \mathbf{I}_M$; we conclude that the distributed filter is mean stable if $|1 - \mu_i \sigma_{u,i}^2| < 1$, $i = 1, \dots, N$, i.e., if

$$0 < \mu_i < \frac{2}{\sigma_{u,i}^2} \quad i = 1, \dots, N,$$

which provides the stability condition of the proposed algorithm.

VI. EXPERIMENTS

In this section, we demonstrate the significant reduction in the communication load achieved by our algorithms while providing equal performance with respect to the state of the art. For the simulations, we consider a sample network consisting of $N = 10$ nodes, where each node makes a scalar observation via the linear model (1). The regressor standard deviations $\sigma_{u,i}$ are chosen randomly from the interval $(0.1, 0.3)$. The observation noise is generated from a Normal distribution with variance $\sigma_v^2 = 0.01$. The unknown vector parameter \mathbf{w}_o with $M = 10$ components is randomly chosen from a Normal distribution, and normalized to unit energy. This randomization is repeated one more time within the course of the simulation to observe how well the algorithm is able to track sudden changes in the unknown parameter. We use the Metropolis rule to generate the network matrix \mathbf{P} using

$$P_{i,j} = \begin{cases} \frac{2}{M^2 \max(N_i, N_j)} & \text{if } i \neq j \text{ are linked,} \\ 0 & \text{for } i \text{ and } j \text{ not linked,} \\ 1 - \sum_{j \in \mathcal{N}_i \setminus \{i\}} \lambda_{i,j} & \text{for } i = j \end{cases}$$

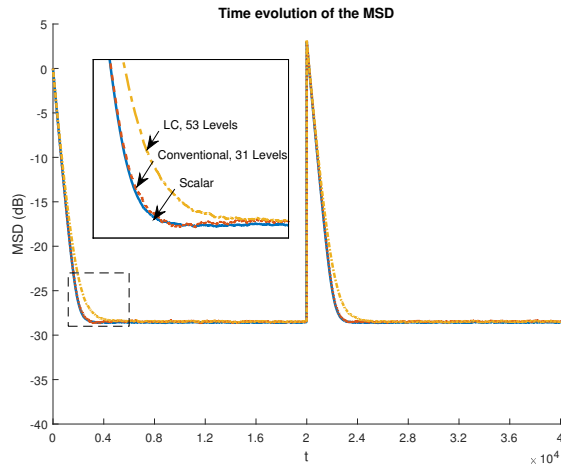


Fig. 3. MSD Performance of the proposed algorithm, represented with the label 'LC'.

We configure the nodes such that they cycle through the entries of the intermediary estimates in a round-robin fashion, and exchange only one out of M components [11].

We compare the proposed algorithm with [11] and demonstrate that our algorithm significantly enhances the efficiency of the adaptive network in terms of the incurred communication cost. In Figure 4, the mean-square deviation (MSD) performance, given by $E\|\tilde{\mathbf{w}}_t\|^2$ of the proposed algorithm is demonstrated, where as a reference, we have considered [11] with an adaptive Lloyd-Max quantizer, and the no-quantization (scalar) implementation of the system. The simulations use a value of $\mu = 0.05$. Figure 5 demonstrates the substantial enhancement in the communication efficiency achieved by the proposed algorithm, in terms of the total number of bits exchanged between the nodes across the entire adaptive network. In particular, we see that the proposed algorithm provides three orders of magnitude improvement over the reference implementation in terms of the communication load on the network, while almost exactly matching it in terms of the steady-state global mean-square deviation, the speed of convergence and the tracking performance. We stress further that we achieve this improvement with relatively little complexity since we have shown that using a simple non-adaptive quantizer is sufficient to realize the improvements.

VII. CONCLUSION

We introduced an event-triggered distributed estimation algorithm with level-crossing quantization for distributed applications, where an unknown parameter is cooperatively learned by a group of nodes in an adaptive network. We proposed a diffusion-LMS algorithm where at each time instant, a node initiates communication with its neighbours only if the parameter to be communicated goes through a level crossing, which is signified by a single bit that indicates the direction of the level crossing. Consequently, the proposed algorithm required data transfers between the nodes that are much more sparse across time, as compared to a continuous stream of information at each instant. This translated into a much diminished load on available communication resources, which

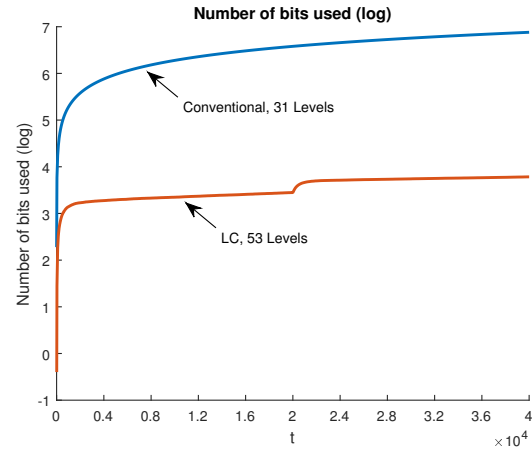


Fig. 4. Time evolution of the total number of bits transmitted on the network.

is of crucial importance in applications such as big data, where these resources are constrained, set against the sheer volume of the data. By theoretical analysis and simulations, we showed that the proposed algorithm is convergent in the mean sense, and we demonstrated that it achieves up to three orders-of-magnitude improvement in the communication load imposed on the network.

ACKNOWLEDGMENT

This work is supported in part by TUBITAK, Contract no: 113E517.

REFERENCES

- [1] F. S. Cattivelli and A. H. Sayed, "Diffusion lms strategies for distributed estimation," *IEEE Transactions on Signal Processing*, vol. 58, no. 3, pp. 1035–1048, 2010.
- [2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP '01). 2001 IEEE International Conference on*, vol. 4, 2001, pp. 2033–2036 vol.4.
- [3] D. Li, K. Wong, Y. H. Hu, and A. Sayeed, "Detection, classification, and tracking of targets," *Signal Processing Magazine, IEEE*, vol. 19, no. 2, pp. 17–29, Mar 2002.
- [4] J.-J. Xiao, A. Ribeiro, Z.-Q. Luo, and G. Giannakis, "Distributed compression-estimation using wireless sensor networks," *Signal Processing Magazine, IEEE*, vol. 23, no. 4, pp. 27–41, July 2006.
- [5] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3122–3136, 2008.
- [6] Z. Xiaochuan and A. Sayed, "Single-link diffusion strategies over adaptive networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, March 2012, pp. 3749–3752.
- [7] J. W. Mark and T. Todd, "A nonuniform sampling approach to data compression," *Communications, IEEE Transactions on*, vol. 29, no. 1, pp. 24–32, Jan 1981.
- [8] C. Lopes and A. Sayed, "Diffusion adaptive networks with changing topologies," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, March 2008, pp. 3285–3288.
- [9] N. Takahashi and I. Yamada, "Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 3518–3521.
- [10] T. G. M. O. Sayin, N. D. Vanli and S. S. Kozat, "Communication efficient channel estimation over distributed networks," in *IEEE Global Conference on Signal and Information Processing, GlobalSIP*, 2014.
- [11] R. Arablouei, S. Werner, Y. F. Huang, and K. Dogancay, "Distributed least mean square estimation with partial diffusion," *IEEE Transactions on Signal Processing*, vol. 62, no. 2, pp. 472–483, 2014.
- [12] A. H. Sayed, *Fundamentals of Adaptive Filtering*. John Wiley and Sons, 2003.